小型在线商城的数据库应用设计报告

16200422 尉文硕 材料科学与工程学院

目录

一、需求分析

1.任务概述

(1) 背景与目的

小型在线商城是一种在互联网上开展电子商务的平台，其基本功能为在线购物，其的业务管理系统针对商品出入、顾客购买、货物供应等信息处理工作来开发，需要根据用户要求实现对商品、顾客、雇员及供应商信息等的管理功能。

本设计目的在于提供电商常用功能模块的数据库设计与常见问题的数据库解决方案。

具体需求如下：

-顾客和商家在首次使用本系统的购物功能前，必须先注册账号。在注册页面填写手机号码和密码等个人信息，完成注册后，系统将保存顾客或商家信息。

-顾客可以浏览商品，顾客登录系统后，可以向购物车中添加、删除商品，修改商品数量；可以购买商品，从购物车中选择一件或多件商品后提交订单；可以查询自己的订单。

-商家成功登录系统后，可以添加、删除商品，可以修改商品信息；可以通过关键字搜索指定的商品信息；可以对订单进行查询。

（2）用户特征

数据库系统终端分为前台和后台。前台面向访问网站的用户，用于访问网站选购商品；后台面向经营者，用于处理货物、顾客、雇员、供应商相关数据。

（3）运行条件与限制

硬件环境：Apple MacBook M1

软件环境：macOS 13.2
Mysql 8.0.30
MySQLWorkbench
Visual Studio Code 1.75.0
Python 3.9.10

2.功能分析

（1）用例描述

用例名：添加购物车商品
简述：顾客有购买商品的意图，但是觉得需要考虑时，可执行添加购物车商品操作。
参与者：消费者
前置条件：顾客必须登录成功。
细节：在主页面，用户点击商品，进入商品介绍页面；在商品介绍页面，用户点击加入购物车，进入商品信息选择页面；物品信息选择页面，用户选择商品信息，点击确认，系统自动将物品加入购物车。
后置条件：物品信息选择页面，商品信息必须填写完全；购物车的商品数量不能超出特定值。
例外：物品信息选择页面，当商品信息不完全，用户点击确认时系统提示商品信息未填写完全，当购物车的商品数量已达特定值，用户点击确认时系统提示购物车已满。

用例名：上架商品
简述：商家想上架新的商品，扩大经营的时候，可使用上架商品操作。
参与者：商家
前置条件：商家必须登录成功。
细节：在主页面，用户点击商品管理进入商品管理界面，用户点击上架商品，进入上架商品页面；在上架商品页面，用户填写完商品的信息，点击确定，即可上架商品。
后置条件：商品信息填写完全

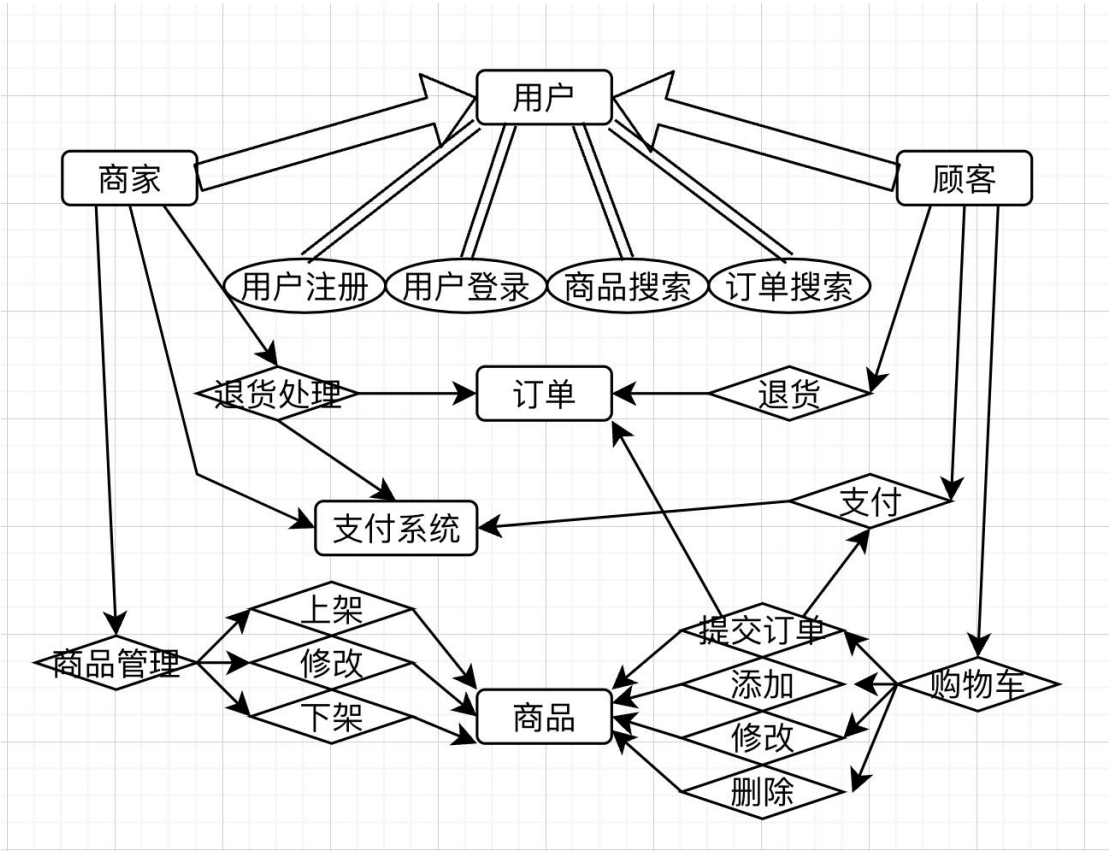
用例名：顾客退货
简述：顾客不满意商品需要退款，可使用退货处理操作。
参与者：商家、顾客
前置条件：顾客、商家必须登录成功。

细节：在主页面，用户点击退货处理进入退货管理界面，用户输入退货处理订单号，选择确认退货。
后置条件：商家选择退货处理，确认退货，即可退货

（2）各模块功能需求

i.主页面模块
　　用户注册：用户可对自身信息进行录入
　　用户登录：用户可随时登陆系统
　　搜索商品：用户通过搜索，浏览商品信息
　　订单维护：用户通过搜索，浏览订单信息
ii.商家模块
　　商品管理：商家通过商品管理，实现商品入库、上架、下架和商品信息修改
　　退货处理：商家处理退货信息
iii.顾客模块
　　购物车管理：顾客将想要的商品添加进购物车
　　退货：顾客退货
　　支付：顾客购买商品生成订单并支付

3.系统体系结构



上图描述了小型在线商城业务系统中的基本功能实现。

全过程系统结构清晰明了，简单易懂，条理十分清楚，具有很强的可操作性。

4.性能要求

系统需要容错性好，能够对各种误操作具有足够的鲁棒性和容错性。例如，注销用户身份时，需要再三确认是否进行该次操作。在提交订单支付或者退款过程中，添加相关提示界面，防止重复支付或者支付失败。

系统界面需要人性化，能够进行人机友好交互，添加合适的文字指引与图片说明，方便理解与操作。

系统需要有性能良好的硬件存储设备支撑，从而能够存储足够大的商品流通信息。

5.其他要求

（1）管理员权限要求：

要求管理员对顾客端所有信息变更具有审核和修改权限。

（2）数据历史记录要求：

要求自动保存每次数据变更至历史记录，同步至本地。避免因为意外丢失数据，造成管理系统崩溃。同时避免因为误操作损坏数据、引入脏数据等，造成正常的数据变动出现故障。

6.可行性分析

可行性分析对系统的开发至关重要，可以大幅减少不必要的损失，保证系统开发的顺利进行。因此要对系统进行技术可行性、经济可行性、营运可行性三方面的系统可行性分析：

（1）技术可行性

计算机技术发展迅猛，高速度大容量的电脑已成为许多公司日常工作必不可少的设备，推动办公自动化的软件不断涌现，微机的普及为该系统的开发奠定了坚实的基础。数据库软件采用著名的开源软件 mysql，开发接口好，适合非计算机人员使用。

（2）经济可行性

新系统的开发由于使用开源软件，不需要额外增加设备和软件购置费，只需要少量的软件开发费、管理和维护费用，且人员培训过程简单费用不多。另一方面，新系统的开发可以较好地解决商场因业务繁杂而造成的处理效率低，出错率偏高的局面，并可以及时了解各项业务的进展情况，为及时调整经营决策提供可靠的数据支持，从而提高经济效益，同时还可以减少人工劳动、提高工作效率、扩大业务量和竞争能力。

（3）营运可行性

计算机在现代社会得到越来越普及的应用，人们越发习惯计算机的思维方式和使用模式。建立一个在线商场管理系统数据库应用，使用界面良好，易于操作。

二、实体概念设计和关系设计

1. 设计实体

实体	说明	组成
会员	记录会员的基本信息	会员编号、姓名、密码、电话、地址、积分
商品	记录销售商提供的商品信息	商品编号、名称、价格、数量、销售商
订单	记录会员的购物信息	会员编号、商品编号、订单编号、订单日期、最后总价
购物车	存放会员需要购买的商品	会员编号、商品编号、购物车编号、商品数量
管理员	记录管理员的基本信息	管理员编号、管理员姓名、密码、联系电话
存货	记录仓库商品储存信息	商品编号、名称、价格、库存数量、销售商、进价

2. 属性设计

(1)注册会员信息管理部分。

要求:

a 可以查看注册会员信息。

b.可以对注册会员信息进行添加及删除的操作。

(2)商品信息管理部分。

要求:

a 可以浏览商品信息。

b.可以对商品信息进行维护,包括添加及删除的操作。

(3)购物车信息管理部分。

要求:

a 可以浏览购物车信息。

b.可以对购物车信息进行维护,包括添加及删除的操作。

(4)订单信息管理部分。

要求:

a 可以浏览订单信息。

b.可以对订单信息进行维护，包括添加及删除的操作。

(5)管理员信息管理部分。

要求:

a 可以浏览管理员信息。

b.对管理员信息进行维护，包括添加及删除的操作。

(6)存货信息管理部分。

要求:

a 显示当前存货信息。

b.对存货信息进行维护操作，包括添加及删除的操作。

3.关系分析

-用户表与订单表：一对多（订单表加外键指向用户表的用户 ID）

一个用户可以有多个订单，所以是一对多

-雇员表与订单表：一对多（订单表加外键指向雇员表的用户 ID）

一个雇员可以处理多个订单，所以是一对多

-雇员表与库存表：一对多（库存表加外键指向雇员表的用户 ID）

一个雇员可以处理多个库存，所以是一对多

-库存表与商品表：一对一（商品表中加外键指向库存表的 ID）

一个商品中对应一个库存商品，所以是一对一

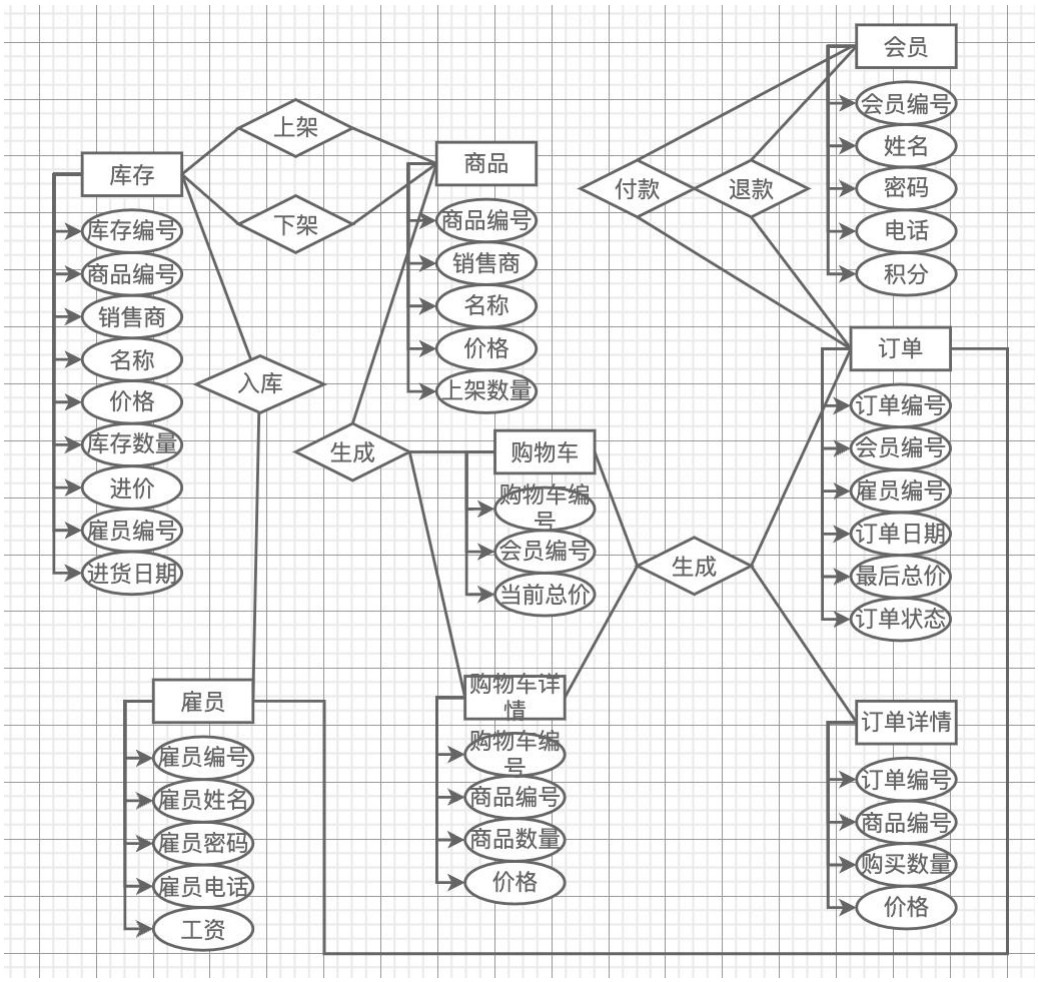
-订单表与商品表：多对多（中间表：创建一张两列的表分别指向订单编号和商品编号。）

一个订单里可以有多个商品，一个商品也可以在多个订单里，所以是多对多

-购物车表与商品表：多对多（中间表：创建一张两列的表分别指向购物车编号和商品编号。）

一个购物车里可以有多个商品，一个商品也可以在多个购物车里，所以是多对多

4.E-R 图



三、逻辑实现

1. 确定字段与关系表

会员信息表:

列名	数据类型	长度	说明	是否为空
会员编号	整型	10	会员的编号	主键,非空
姓名	文本型	20	会员的姓名	非空
密码	字符型	20	会员的密码	非空
电话	文本型	12	会员的电话	
积分	整型	10	会员的积分	

商品信息表:

列名	数据类型	长度	说明	是否为空

商品编号	整型	10	商品的编号	主键,非空
销售商	文本型	20	商品的销售商	非空
名称	文本型	20	商品的名称	非空
价格	字符型	6	商品的价格	非空
上架数量	整型	10	商品的数量	非空

购物车信息表:

列名	数据类型	长度	说明	是否为空
购物车编号	整型	10	购物车的编号	主键,非空
会员编号	整型	10	会员的编号	非空
当前总价	字符型	10	购买商品的总价格	非空

订单信息表:

列名	数据类型	长度	说明	是否为空
订单编号	整型	10	购物时生成的订单	主键,非空
会员编号	整型	10	会员的编号	非空
订单日期	时间型	10	购买商品的时间	非空
最后总价	字符型	10	购买商品的总价格	非空
订单状态	字符型	1	订单付款/退款（y/n）	非空
雇员编号	整型	10	处理订单的雇员	非空

雇员信息表:

列名	数据类型	长度	说明	是否为空
雇员编号	整型	10	雇员的编号	主键,非空
雇员姓名	文本型	20	雇员的姓名	非空
雇员密码	字符型	20	雇员的密码	非空
雇员电话	字符型	12	雇员的联系电话	非空
工资	字符型	10	雇员的工资	非空

库存信息表:

列名	数据类型	长度	说明	是否为空
库存编号	整型	10	库存的编号	主键，非空
商品编号	整型	10	商品的编号	非空
销售商	文本型	20	商品的销售商	非空
名称	文本型	20	商品的名称	非空
价格	字符型	10	商品的价格	非空
库存数量	整型	10	商品的数量	非空
进价	字符型	10	商品的进价	非空
雇员编号	整型	10	处理库存的雇员	非空

			的编号	
进货日期	时间型	10	进货的时间	非空

2. 确立中间表

订单详情表：

订单编号	整型	10	购物时生成的订单	非空
商品编号	整型	10	商品的编号	非空
购买数量	整型	10	购买商品的数量	非空
价格	字符型	10	商品的价格	非空

购物车详情表：

商品编号	整型	10	商品的编号	非空
商品数量	整型	10	购买商品的数量	非空
购物车编号	整型	10	购物车的编号	非空
价格	字符型	10	商品的价格	非空

四、物理实现

1. 建立一个数据库

```
create database shop;
```

2. 建立关系表与字段序列

```
use shop;

create table  雇员信息表(
雇员编号      int        not null primary key auto_increment,
雇员姓名      text             not null,
雇员密码                 char(20)      not null,
雇员电话          text             null,
工资                decimal(8,2)              null
)ENGINE=InnoDB auto_increment 10000 DEFAULT CHARSET=gb2312;#utf8mb4;

create table  会员信息表(
```

```
会员编号      int          not null primary key auto_increment,
姓名          text                  not null,
密码          char(20)              not null,
电话          text                  null,
积分          int                   null
)ENGINE=InnoDB auto_increment 100000 DEFAULT CHARSET=gb2312;#utf8mb4;

create table  商品信息表(
商品编号      int       not null primary key,
名称          text              not null,
价格          decimal(8,2)              null,
上架数量      int       not null,
销售商        text              null
#INDEX  商品名称索引(名称),UNIQUE INDEX  商品编号索引(商品编号)
)ENGINE=InnoDB DEFAULT CHARSET=gb2312;#utf8mb4;

create table  购物车信息表(
购物车编号    int    not null primary key auto_increment,
会员编号      int    not null,
当前总价    decimal(8,2)     not null
)ENGINE=InnoDB auto_increment 1000000 DEFAULT CHARSET=gb2312;#utf8mb4;

create table  订单信息表(
订单编号    int          not null    primary key auto_increment,
会员编号    int        not null,
订单日期    datetime         not null,
最后总价    decimal(8,2)     not null,
订单状态     char(1)          not null default 'y'
)ENGINE=InnoDB auto_increment 1000000 DEFAULT CHARSET=gb2312;#utf8mb4;

create table  库存信息表(
库存编号      int          not null primary key auto_increment,
商品编号      int       not null,
库存数量      int       not null,
名称          text              not null,
价格          decimal(8,2)              null,
销售商        text              null,
进价          decimal(8,2)              null,
雇员编号    int       not null,
进货日期    datetime         not null
)ENGINE=InnoDB auto_increment 1000000 DEFAULT CHARSET=gb2312;#utf8mb4;

create table  订单详情表(
订单编号    int          not null,
```

```
商品编号      int        not null,
购买数量      int        not null,
价格          decimal(8,2)          null
)ENGINE=InnoDB DEFAULT CHARSET=gb2312;#utf8mb4;

create table  购物车详情表(
购物车编号      int          not null,
商品编号      int        not null,
商品数量      int        not null,
价格          decimal(8,2)          null
)ENGINE=InnoDB DEFAULT CHARSET=gb2312;#utf8mb4;
```

3. 创建索引

```
#创建索引
CREATE UNIQUE INDEX  会员编号索引  ON  会员信息表(会员编号);
CREATE UNIQUE INDEX  商品编号索引  ON  商品信息表(商品编号);
CREATE UNIQUE INDEX  雇员编号索引  ON  雇员信息表(雇员编号);
```

4. 建立各表之间的联系

```
alter table  订单信息表
add constraint FK_订单信息表_REFRENCE_会员信息表  foreign key (会员编号)
references  会员信息表(会员编号);

alter table  订单详情表
add constraint FK_订单详情表_REFRENCE_商品信息表  foreign key (商品编号)
references  商品信息表(商品编号);

alter table  订单详情表
add constraint FK_订单详情表_REFRENCE_订单信息表  foreign key (订单编号)
references  订单信息表(订单编号);

alter table  库存信息表
add constraint FK_库存信息表_REFRENCE_商品信息表  foreign key (商品编号)
references  商品信息表(商品编号);

alter table  库存信息表
add constraint FK_库存信息表_REFRENCE_雇员信息表  foreign key (雇员编号)
references  雇员信息表(雇员编号);

alter table  购物车信息表
```

add constraint FK_购物车信息表_REFRENCE_会员信息表 foreign key (会员编号)
references 会员信息表 (会员编号);

alter table 购物车详情表
add constraint FK_购物车详情表_REFRENCE_商品信息表 foreign key (商品编号)
references 商品信息表(商品编号);

alter table 购物车详情表
add constraint FK_购物车详情表_REFRENCE_购物车信息表 foreign key (购物车编号)
references 购物车信息表(购物车编号);

```
mysql> desc 雇员信息表;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| 雇员编号     | int          | NO   | PRI | NULL    | auto_increment |
| 雇员姓名     | text         | NO   |     | NULL    |                |
| 雇员密码     | char(20)     | NO   |     | NULL    |                |
| 雇员电话     | text         | YES  |     | NULL    |                |
| 工资         | decimal(8,2) | YES  |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
5 rows in set (0.01 sec)

mysql> desc 会员信息表;
+--------------+----------+------+-----+---------+----------------+
| Field        | Type     | Null | Key | Default | Extra          |
+--------------+----------+------+-----+---------+----------------+
| 会员编号     | int      | NO   | PRI | NULL    | auto_increment |
| 姓名         | text     | NO   |     | NULL    |                |
| 密码         | char(20) | NO   |     | NULL    |                |
| 电话         | text     | YES  |     | NULL    |                |
| 积分         | int      | YES  |     | NULL    |                |
+--------------+----------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> desc 商品信息表;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| 商品编号     | int          | NO   | PRI | NULL    |       |
| 名称         | text         | NO   |     | NULL    |       |
| 价格         | decimal(8,2) | YES  |     | NULL    |       |
| 上架数量     | int          | NO   |     | NULL    |       |
| 销售商       | text         | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> desc 购物车信息表;
+----------------+--------------+------+-----+---------+----------------+
| Field          | Type         | Null | Key | Default | Extra          |
+----------------+--------------+------+-----+---------+----------------+
| 购物车编号     | int          | NO   | PRI | NULL    | auto_increment |
| 会员编号       | int          | NO   | MUL | NULL    |                |
| 当前总价       | decimal(8,2) | NO   |     | NULL    |                |
+----------------+--------------+------+-----+---------+----------------+
3 rows in set (0.01 sec)
```

```
mysql> desc 订单信息表;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| 订单编号     | int          | NO   | PRI | NULL    | auto_increment |
| 会员编号     | int          | NO   | MUL | NULL    |                |
| 订单日期     | datetime     | NO   |     | NULL    |                |
| 最后总价     | decimal(8,2) | NO   |     | NULL    |                |
| 订单状态     | char(1)      | NO   |     | y       |                |
+--------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> desc 库存信息表;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| 库存编号     | int          | NO   | PRI | NULL    | auto_increment |
| 商品编号     | int          | NO   | MUL | NULL    |                |
| 库存数量     | int          | NO   |     | NULL    |                |
| 名称         | text         | NO   |     | NULL    |                |
| 价格         | decimal(8,2) | YES  |     | NULL    |                |
| 销售商       | text         | YES  |     | NULL    |                |
| 进价         | decimal(8,2) | YES  |     | NULL    |                |
| 雇员编号     | int          | NO   | MUL | NULL    |                |
| 进货日期     | datetime     | NO   |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
9 rows in set (0.00 sec)

mysql> desc 订单详情表;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| 订单编号     | int          | NO   | MUL | NULL    |       |
| 商品编号     | int          | NO   | MUL | NULL    |       |
| 购买数量     | int          | NO   |     | NULL    |       |
| 价格         | decimal(8,2) | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> desc 购物车详情表;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| 购物车编号     | int          | NO   | MUL | NULL    |       |
| 商品编号       | int          | NO   | MUL | NULL    |       |
| 商品数量       | int          | NO   |     | NULL    |       |
| 价格           | decimal(8,2) | YES  |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

5. 设计和实现视图

（1）会员订单视图

drop view if exists 会员订单视图;
create view 会员订单视图(会员编号,姓名,订单编号,订单状态,商品编号,购买数量,订单日期)
as select 订单信息表.会员编号,会员信息表.姓名,订单信息表.订单编号,订单信息表.订单状

态,订单详情表.商品编号,订单详情表.购买数量,订单信息表.订单日期
from 订单信息表,会员信息表,订单详情表 where 订单信息表.订单状态='y'
and 订单信息表.会员编号=会员信息表.会员编号 and 订单信息表.订单编号=订单详情表.订单编号;

```
mysql> desc 会员订单视图;
+--------------+----------+------+-----+---------+-------+
| Field        | Type     | Null | Key | Default | Extra |
+--------------+----------+------+-----+---------+-------+
| 会员编号     | char(10) | NO   |     | NULL    |       |
| 姓名         | text     | NO   |     | NULL    |       |
| 订单编号     | char(10) | NO   |     | NULL    |       |
| 订单状态     | char(1)  | NO   |     | y       |       |
| 商品编号     | char(10) | NO   |     | NULL    |       |
| 购买数量     | int      | NO   |     | NULL    |       |
| 订单日期     | datetime | NO   |     | NULL    |       |
+--------------+----------+------+-----+---------+-------+
7 rows in set (0.01 sec)
```

（2）会员购物车视图

drop view if exists 会员购物车视图;
create view 会员购物车视图(会员编号,姓名,购物车编号,商品编号,商品数量)
as select 购物车信息表.会员编号,会员信息表.姓名,购物车信息表.购物车编号,购物车详情表.商品编号,购物车详情表.商品数量
from 购物车信息表,会员信息表,购物车详情表
where 购物车信息表.会员编号=会员信息表.会员编号 and 购物车信息表.购物车编号=购物车详情表.购物车编号;

```
mysql> desc 会员购物车视图;
+----------------+------+------+-----+---------+-------+
| Field          | Type | Null | Key | Default | Extra |
+----------------+------+------+-----+---------+-------+
| 会员编号       | int  | NO   |     | NULL    |       |
| 姓名           | text | NO   |     | NULL    |       |
| 购物车编号     | int  | NO   |     | 0       |       |
| 商品编号       | int  | NO   |     | NULL    |       |
| 商品数量       | int  | NO   |     | NULL    |       |
+----------------+------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

（3）销售情况视图

drop view if exists 销售情况视图;
create view 销售情况视图(商品编号,购买数量,订单日期)
as select 订单详情表.商品编号,订单详情表.购买数量,订单信息表.订单日期
from 订单信息表,订单详情表 where 订单信息表.订单状态='y' and 订单信息表.订单编号=订单详情表.订单编号;

```
mysql> desc 销售情况视图;
+--------------+----------+------+-----+---------+-------+
| Field        | Type     | Null | Key | Default | Extra |
+--------------+----------+------+-----+---------+-------+
| 商品编号     | char(10) | NO   |     | NULL    |       |
| 购买数量     | int      | NO   |     | NULL    |       |
| 订单日期     | datetime | NO   |     | NULL    |       |
+--------------+----------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

6.实现存储过程

（1）完成付款操作

```
use shop;

DROP PROCEDURE if exists 付款;

DELIMITER $$
CREATE PROCEDURE 付款(IN 购物车编号参数 int,IN 雇员编号参数 int)
BEGIN
    DECLARE 会员编号参数 int;
    DECLARE 商品编号参数 int;
    DECLARE 订单编号参数 int;
    DECLARE 商品数量参数 int;
    DECLARE 订单日期参数 DATETIME;
    DECLARE 商品价格参数 DECIMAL(8,2);
    DECLARE 当前总价参数 DECIMAL(8,2);

    SELECT 会员编号, 当前总价 INTO 会员编号参数, 当前总价参数 FROM 购物车信息表 WHERE 购物车编号 = 购物车编号参数;
    SELECT 商品编号, 商品数量 INTO 商品编号参数, 商品数量参数 FROM 购物车详情表 WHERE 购物车编号 = 购物车编号参数;

    #从购物车创立订单
    INSERT INTO 订单信息表 VALUES(订单编号参数=null,会员编号参数,雇员编号参数,CURRENT_TIMESTAMP(),当前总价参数,'y');
    INSERT INTO 订单详情表 VALUES(订单编号参数,商品编号参数,商品数量参数,商品价格参数);

    # 更新商品库存
    UPDATE 商品信息表 SET 上架数量 = 上架数量 -商品数量参数 WHERE 商品编号 = 商品编号参数;
    # 更新会员积分
    UPDATE 会员信息表 SET 积分 = 积分 + (当前总价参数 *0.1) WHERE 会员编号 =
```

```
会员编号参数;
END$$
DELIMITER ;
```

（2）完成退款操作

```
use shop;

DROP PROCEDURE if exists 退款;

DELIMITER $$
CREATE PROCEDURE 退款(IN 订单编号参数 int)
BEGIN
    DECLARE 商品编号参数 int;
    DECLARE 购买数量参数 int;
    DECLARE 最后总价参数 DECIMAL(8,2);
    DECLARE 商品价格参数 DECIMAL(8,2);
    DECLARE 会员编号参数 int;

    SELECT 价格, 商品编号, 购买数量 INTO 商品价格参数, 商品编号参数, 购买数量参数 FROM 订单详情表 WHERE 订单编号 = 订单编号参数;
    SELECT 会员编号,最后总价 INTO 会员编号参数,最后总价参数 FROM 订单信息表 WHERE 订单编号 = 订单编号参数;

    # 更新商品库存
    UPDATE 商品信息表 SET 上架数量 = 上架数量 + 购买数量参数 WHERE 商品编号 = 商品编号参数;
    # 更新顾客积分
    UPDATE 会员信息表 SET 积分 = 积分 - (最后总价参数* 0.1) WHERE 会员编号 = 会员编号参数;
    # 更新订单商品处理进度
    UPDATE 订单信息表 SET 订单状态 = 'n',订单日期 = CURRENT_TIMESTAMP() WHERE 订单编号 = 订单编号参数;

END$$
DELIMITER ;
```

（3）完成入库操作

```
use shop;

DROP PROCEDURE if exists 入库;
```

```
DELIMITER $$
CREATE PROCEDURE 入库(IN 库存编号参数 int,IN 商品编号参数 int,IN 销售商参数 TEXT,IN 名称参数 TEXT,IN 价格参数 DECIMAL(8,2),IN 库存数量参数 INT,IN 进价参数 DECIMAL(8,2),IN 雇员编号参数 CHAR(10))
BEGIN
    DECLARE 进货日期参数 datetime;
    declare num int;

    select count(*) into num from 库存信息表 where 商品编号 = 商品编号参数;
    if num=0 then
        INSERT INTO 库存信息表 VALUES(库存编号参数,商品编号参数,库存数量参数,名称参数,价格参数,销售商参数,进价参数,雇员编号参数,CURRENT_TIMESTAMP());
    ELSE
        UPDATE 库存信息表 SET 库存数量 = 库存数量 + 库存数量参数 WHERE 商品编号 = 商品编号参数;
    END IF;

END$$
DELIMITER ;
```

（4）完成上架操作

```
use shop;

DROP PROCEDURE if exists 上架;

DELIMITER $$
CREATE PROCEDURE 上架(IN 商品编号参数 int,IN 上架数量参数 INT)
BEGIN
    DECLARE 名称参数 text;
    DECLARE 销售商参数 text;
    DECLARE 价格参数 DECIMAL(8,2);

    declare num int;
    select count(*) into num from 商品信息表 where 商品编号 = 商品编号参数;

    SELECT 价格, 名称, 销售商 INTO 价格参数, 名称参数, 销售商参数 FROM 库存信息表 WHERE 商品编号 = 商品编号参数;

    if num=0 then
        INSERT INTO 商品信息表 VALUES(商品编号参数,名称参数,价格参数,上架数量参数,销售商参数);
    ELSE
```

```
        UPDATE 商品信息表 SET 上架数量 = 上架数量 + 上架数量参数 WHERE 商品编号 = 商品编号参数;
    END IF;

END$$
DELIMITER ;
```

（5）完成下架操作

```
use shop;

DROP PROCEDURE if exists 下架;

DELIMITER $$
CREATE PROCEDURE 下架(IN 商品编号参数 int)
BEGIN
    DECLARE 上架数量参数 int;
    DECLARE 库存数量参数 int;

    SELECT 上架数量 INTO 上架数量参数 FROM 商品信息表 WHERE 商品编号 = 商品编号参数;
    SELECT 库存数量 INTO 库存数量参数 FROM 库存信息表 WHERE 商品编号 = 商品编号参数;
    UPDATE 商品信息表 SET 上架数量 = 0 WHERE 商品编号 = 商品编号参数;
    UPDATE 库存信息表 SET 库存数量 =库存数量参数+上架数量参数 WHERE 商品编号 = 商品编号参数;
END$$
DELIMITER ;
```

（6）完成加入购物车操作

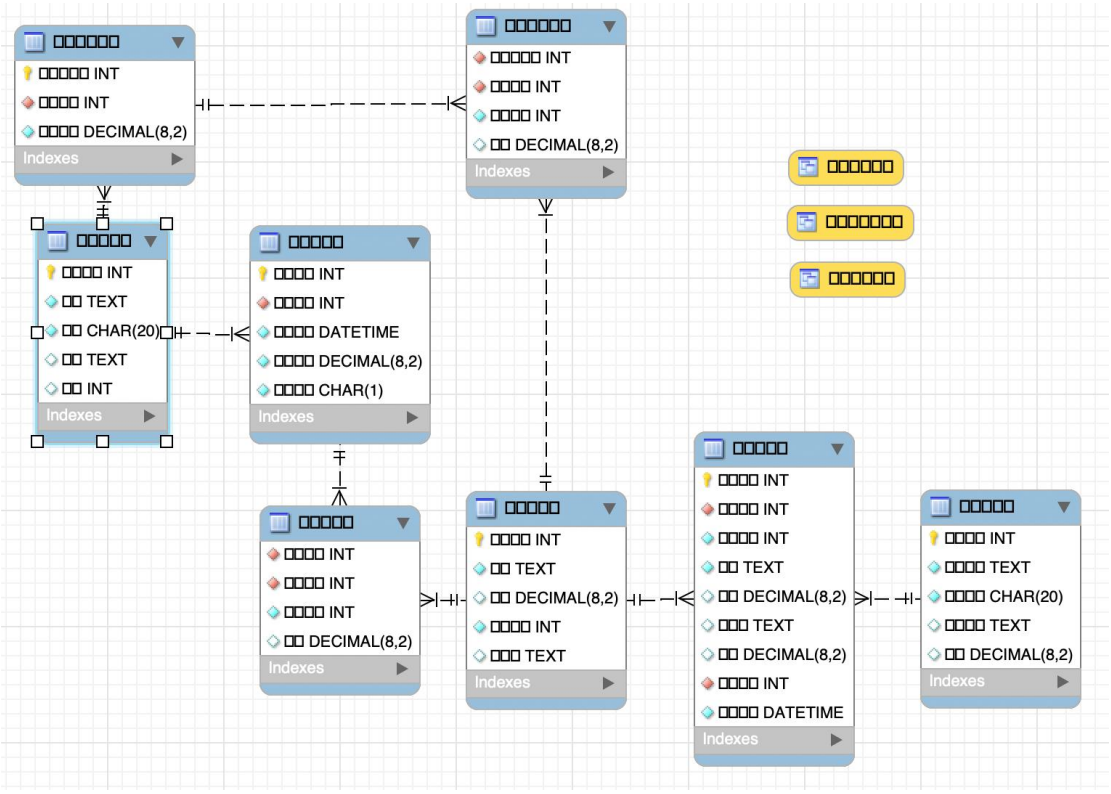
```
use shop;

DROP PROCEDURE if exists 选购;

DELIMITER $$
CREATE PROCEDURE 选购(IN 购物车编号参数 int,IN 会员编号参数 int,IN 商品编号参数 int,IN 商品数量参数 int)
BEGIN
    DECLARE 价格参数 DECIMAL(8,2);
    DECLARE 当前总价参数 DECIMAL(8,2);
```

```
    #SELECT 会员编号 INTO 会员编号参数 FROM 会员信息表 WHERE 会员编号 = 会员编号参数;
    SELECT 价格 INTO 价格参数 FROM 商品信息表 WHERE 商品编号 = 商品编号参数;

    INSERT INTO 购物车详情表 VALUES(购物车编号参数,商品编号参数,商品数量参数,价格参数);

    SELECT SUM(商品数量参数 * 价格参数) as 当前总价参数 FROM 购物车详情表 where 购物车编号=购物车编号参数;

    INSERT INTO 购物车信息表 VALUES(购物车编号参数,会员编号参数,商品数量参数,当前总价参数);
END$$
DELIMITER ;
```

7.物理实现图示

（1）physical schemas diagram

（2）EER 模型视图



五、服务器端存储过程

1. 添加雇员数据：

insert into 雇员信息表(雇员编号,雇员姓名,雇员密码,雇员电话,工资) values (null,'张三','hhjbjkdsjkhbj' ,'110',2500);

insert into 雇员信息表(雇员编号,雇员姓名,雇员密码,雇员电话,工资) values (null,'李四','dbhvhj' ,'120',3000);

2. 从文本文件中导入商品数据

添加商品：

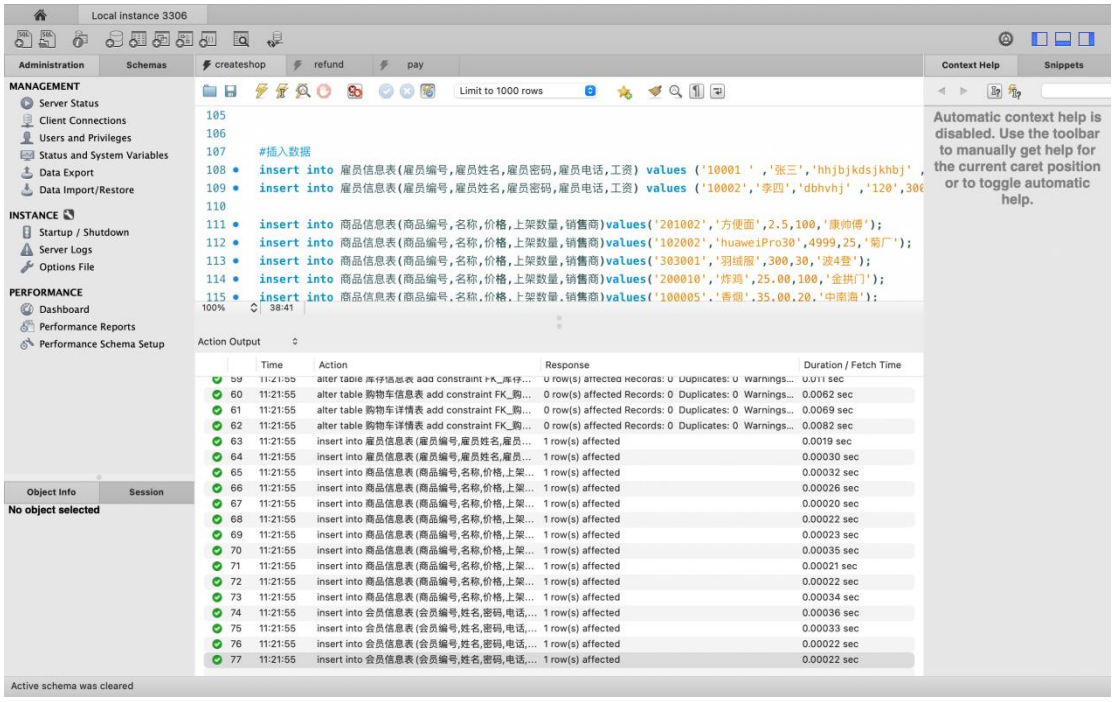
insert into 商品信息表(商品编号,名称,价格,上架数量,销售商)values('201002','方便面',2.5,100,'康帅傅');

insert into 商品信息表(商品编号,名称,价格,上架数量,销售商)values('102002','huaweiPro30',4999,25,'菊厂');

insert into 商品信息表(商品编号,名称,价格,上架数量,销售商)values('303001','羽绒服',300,30,'波 4 登');
insert into 商品信息表(商品编号,名称,价格,上架数量,销售商)values('200010','炸鸡',25.00,100,'金拱门');
insert into 商品信息表(商品编号,名称,价格,上架数量,销售商)values('100005','香烟',35.00,20,'中南海');
insert into 商品信息表(商品编号,名称,价格,上架数量,销售商)values('100010','毛台',400.00,20,'贵酒集团');
insert into 商品信息表(商品编号,名称,价格,上架数量,销售商)values('200011','辣椒酱',12.20,20,'老干爹');
insert into 商品信息表(商品编号,名称,价格,上架数量,销售商)values('400055','手机',1999.00,30,'大米');
insert into 商品信息表(商品编号,名称,价格,上架数量,销售商)values('200014','饼干',5.50,40,'轻食');

3.导入会员数据：

insert into 会员信息表(会员编号,姓名,密码,电话,积分)values(null,'李刚','223456','13029579891',0);
insert into 会员信息表(会员编号,姓名,密码,电话,积分)values(null,'陈明','323456','13019579891',1);
insert into 会员信息表(会员编号,姓名,密码,电话,积分)values(null,'陈志','423456','13719579891',2000);
insert into 会员信息表(会员编号,姓名,密码,电话,积分)values(null,'张力','523456','13729579891',666);



```
mysql> select * from 雇员信息表;
+--------------+--------------+---------------+--------------+---------+
| 雇员编号     | 雇员姓名     | 雇员密码      | 雇员电话     | 工资    |
+--------------+--------------+---------------+--------------+---------+
|        10001 | 张三         | hhjbjkdsjkhbj | 110          | 2500.00 |
|        10002 | 李四         | dbhvhj        | 120          | 3000.00 |
+--------------+--------------+---------------+--------------+---------+
2 rows in set (0.00 sec)

mysql> select * from 会员信息表;
+--------------+--------+--------+-------------+--------+
| 会员编号     | 姓名   | 密码   | 电话        | 积分   |
+--------------+--------+--------+-------------+--------+
|       100000 | 李刚   | 223456 | 13029579891 |      0 |
|       100001 | 陈明   | 323456 | 13019579891 |      1 |
|       100002 | 陈志   | 423456 | 13719579891 |   2000 |
|       100003 | 张力   | 523456 | 13729579891 |    666 |
+--------------+--------+--------+-------------+--------+
4 rows in set (0.00 sec)

mysql> select * from 商品信息表;
+--------------+-------------+---------+--------------+--------------+
| 商品编号     | 名称        | 价格    | 上架数量     | 销售商       |
+--------------+-------------+---------+--------------+--------------+
|       100005 | 香烟        |   35.00 |           20 | 中南海       |
|       100010 | 毛台        |  400.00 |           20 | 贵酒集团     |
|       102002 | huaweiPro30 | 4999.00 |           25 | 菊厂         |
|       200010 | 炸鸡        |   25.00 |          100 | 金拱门       |
|       200011 | 辣椒酱      |   12.20 |           20 | 老干爹       |
|       200014 | 饼干        |    5.50 |           40 | 轻食         |
|       201002 | 方便面      |    2.50 |          100 | 康帅傅       |
|       303001 | 羽绒服      |  300.00 |           30 | 波4登        |
|       400055 | 手机        | 1999.00 |           30 | 大米         |
+--------------+-------------+---------+--------------+--------------+
9 rows in set (0.01 sec)
```

六、客户端编程

1.查询商品信息表示例

```python
import pymysql

connection = pymysql.connect(host='localhost',
   port=3306,
   user='root',
   password='root',
   db='shop',
   charset='utf8')

cursor = connection.cursor()
```

```python
sqlsearch = "SELECT * FROM 商品信息表"
cursor.execute(sqlsearch)

ressearch = cursor.fetchall()
for i in ressearch:
    print(i)

cursor.close()
connection.close()
```

2.执行入库存储进程示例

```python
import pymysql

connection = pymysql.connect(host='localhost',
    port=3306,
    user='root',
    password='root',
    db='shop',
    charset='utf8')

cursor = connection.cursor()

sqlsearch = "select * from 库存信息表"
cursor.execute(sqlsearch)
```

```
ressearch = cursor.fetchall()
for i in ressearch:
    print(i)

#CREATE PROCEDURE 入库(库存编号参数,商品编号参数,销售商参数,名称参数,价格参数,库存数量参数,进价参数,雇员编号参数
sqlstock1 = "call 入库(null,100010,'贵酒集团','毛台
',400,200,200,10000)"
cursor.execute(sqlstock1)
sqlstock2 = "call 入库(null,200014,'饼干','轻食
',5.5,400,3.5,10001)"
cursor.execute(sqlstock2)
sqlsearch = "select * from 库存信息表"

cursor.execute(sqlsearch)
ressearch = cursor.fetchall()
for i in ressearch:
    print(i)

cursor.close()
connection.close()
```

```
sqlsearch = "select * from 库存信息表"
cursor.execute(sqlsearch)
ressearch = cursor.fetchall()
for i in ressearch:
    print(i)
```

[4] ✓ 0.0s Python

```
(1000000, 100010, 100, '毛台', Decimal('400.00'), '贵酒集团', Decimal('200.00'), 10000, datetime.datetime(2023, 2, 12, 19, 56, 9))
```

```
#CREATE PROCEDURE 入库(库存编号参数,商品编号参数,销售商参数,名称参数,价格参数,库存数量参数,进价参数,雇员编号参数
sqlstock1 = "call 入库(null,100010,'贵酒集团','毛台',400,200,200,10000)"
cursor.execute(sqlstock1)
sqlstock2 = "call 入库(null,200014,'饼干','轻食',5.5,400,3.5,10001)"
cursor.execute(sqlstock2)
```

[5] ✓ 0.0s Python

```
1
```

```
sqlsearch = "select * from 库存信息表"
cursor.execute(sqlsearch)
ressearch = cursor.fetchall()
for i in ressearch:
    print(i)
```

[6] ✓ 0.0s Python

```
(1000000, 100010, 300, '毛台', Decimal('400.00'), '贵酒集团', Decimal('200.00'), 10000, datetime.datetime(2023, 2, 12, 19, 56, 9))
(1000001, 200014, 400, '轻食', Decimal('5.50'), '饼干', Decimal('3.50'), 10001, datetime.datetime(2023, 2, 12, 20, 6, 30))
```

七、安全方案设计

1.防止 SQL 注入

SQL 注入：

用户提交带有恶意的数据与 SQL 语句进行字符串方式的拼接，从而影响了 SQL 语句的语义，最后出现数据泄露的现象

SQL 语句参数化：

SQL 语言中的参数使用%s 来占位，这不是 python 中的字符串格式化操作，python 里将 SQL 语句中%s 占位所需要的参数存在一个列表（或元组或字典）中，把参数列表传递给 execute 方法中的第二个参数。

2.备份数据库

数据库的主要作用就是对数据进行保存和维护，所以备份数据是数据库管理中最常用的操作。为了防止数据库意外崩溃或硬件损伤而导致的数据丢失，数据库系统提供了备份和恢复策略。

保证数据安全的最重要的一个措施就是定期的对数据库进行备份。这样即使发生了意外，也会把损失降到最低。

数据库备份是指通过导出数据或者复制表文件的方式来制作数据库的副本。当数据库出现故障或遭到破坏时，将备份的数据库加载到系统，从而使数据库从错误状态恢复到备份时的正确状态。

定期备份数据库能有效保证数据库安全。

示例：`mysqldump -u root -p shop >/Users/yuwenshuo/Documents/shop/shop.sql`

3.启用并检查日志文件

```
mysql> show binary logs;
+---------------+-----------+-----------+
| Log_name      | File_size | Encrypted |
+---------------+-----------+-----------+
| binlog.000012 |       180 | No        |
| binlog.000013 |       324 | No        |
| binlog.000014 |       157 | No        |
| binlog.000015 |     67225 | No        |
| binlog.000016 |    113298 | No        |
| binlog.000017 |      4374 | No        |
+---------------+-----------+-----------+
6 rows in set (0.01 sec)
```

4. 限制访问权限

避免从互联网访问 MySQL 数据库，确保特定主机才拥有访问特权。